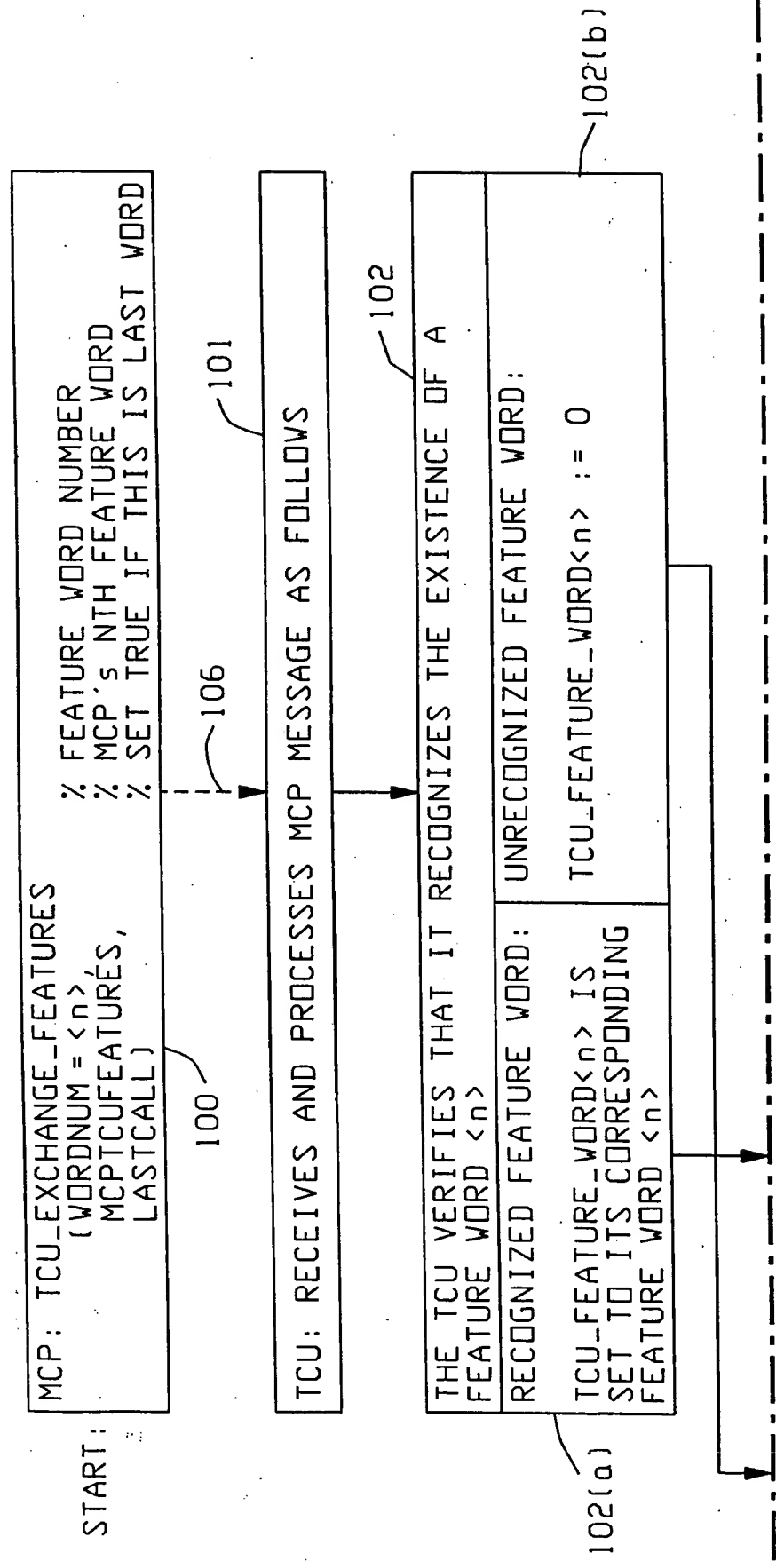




FIG. 3A-1



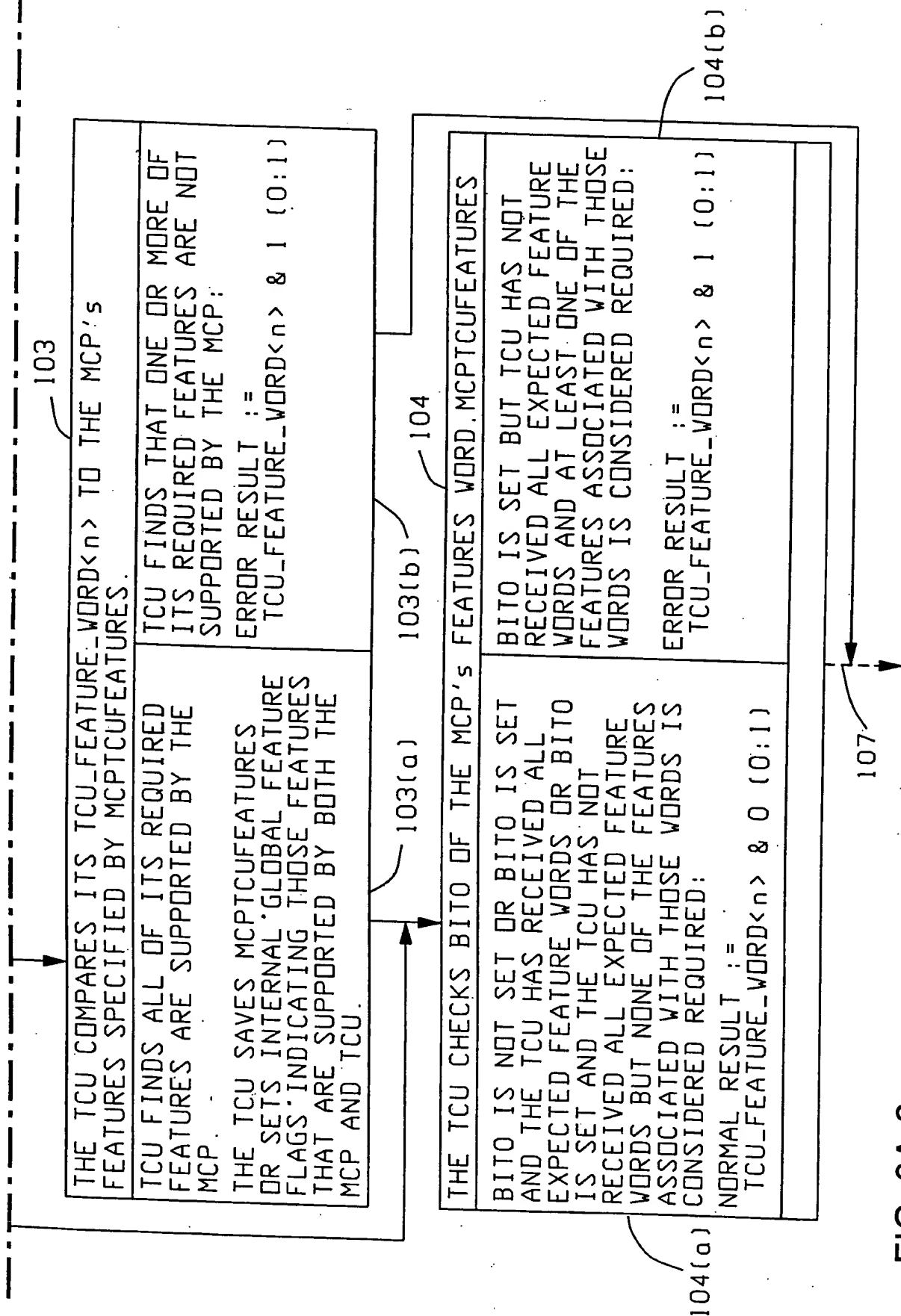
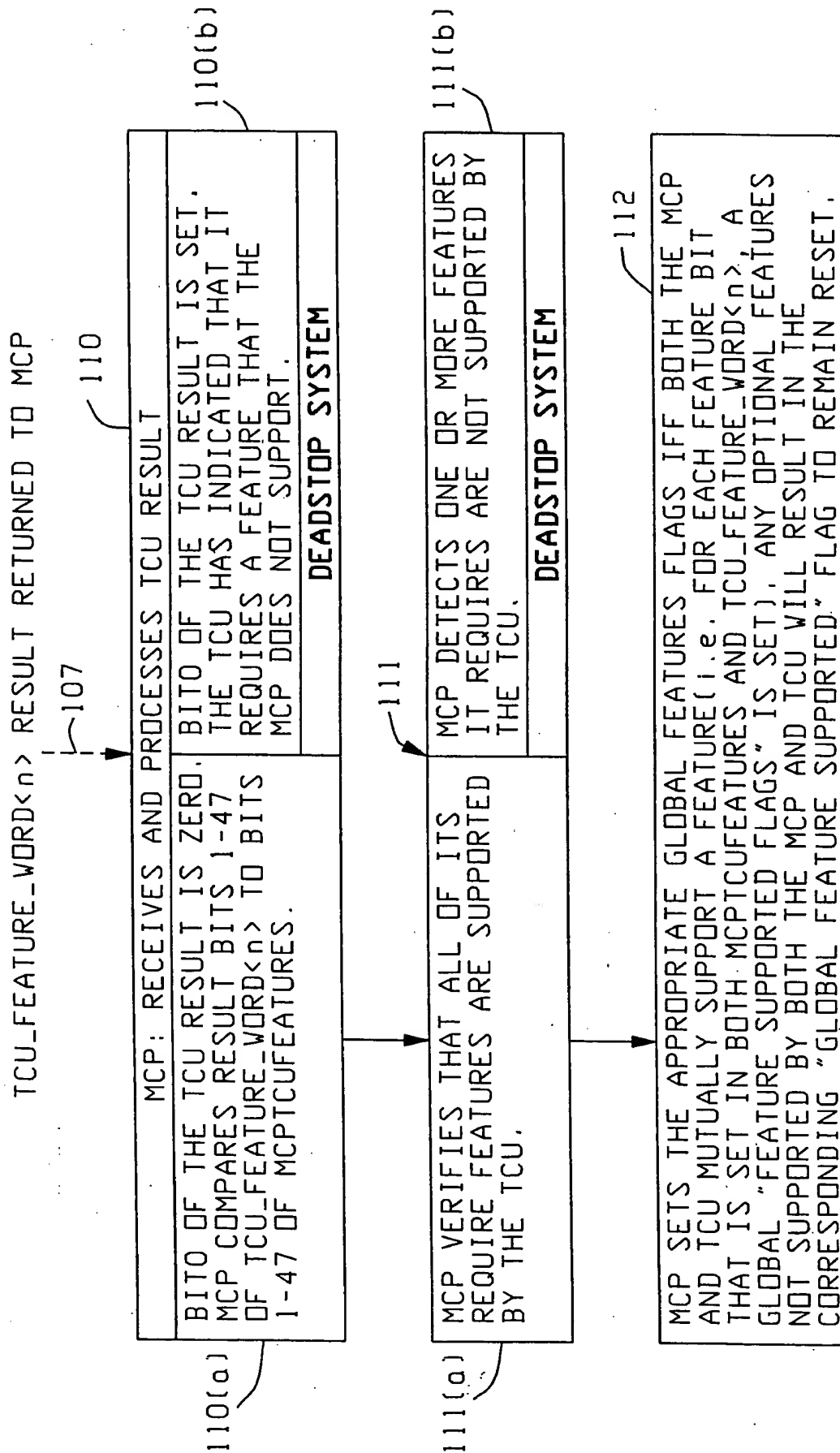


FIG. 3A-2 RETURN TCU_FEATURE_WORD<n> RESULT TO MCP

FIG. 3B-1



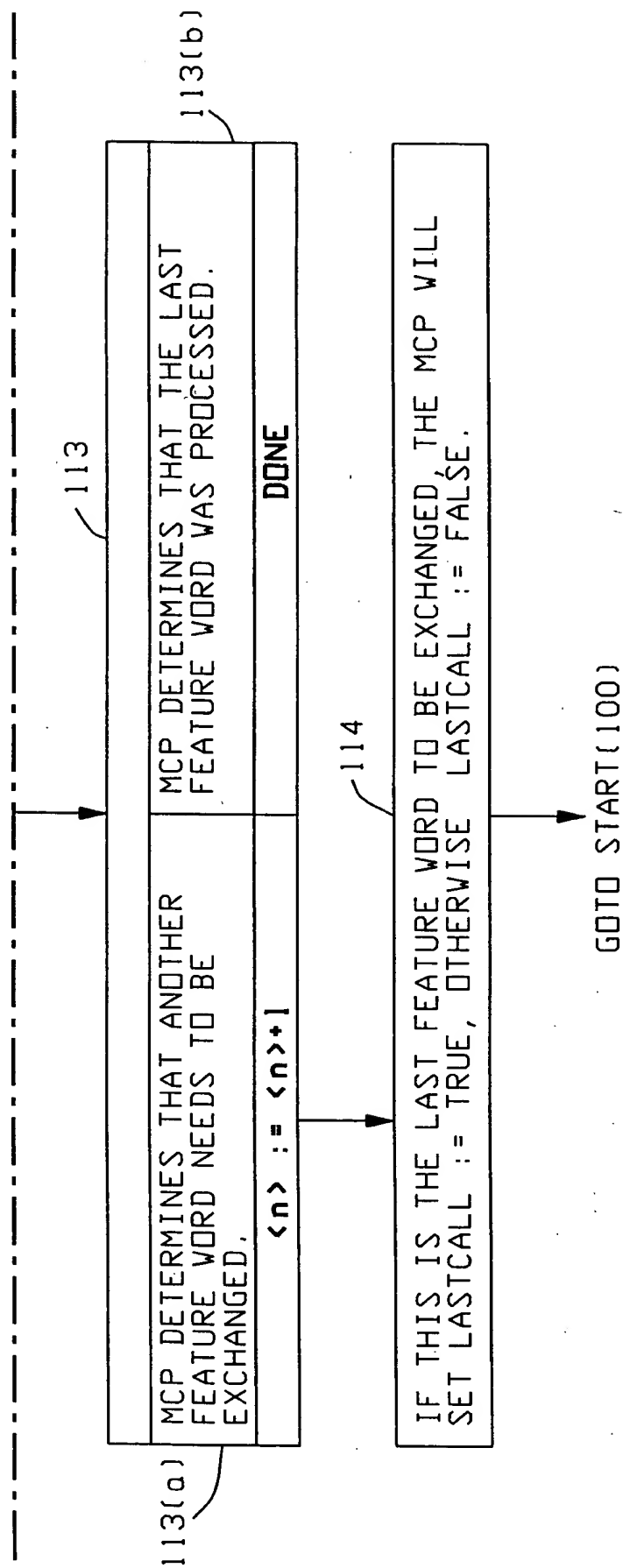


FIG. 3B-2



```
START:      [OS]  WORDNUM_OS := 1;

LOOP:       FEATUREWORD_OS := FEATURES_OS[WORDNUM_OS];      % FEATURES_OS: Array of feature word
                                                    % bit masks supported by OS. This
                                                    % is hardcoded data.

              IF this is the last feature word THEN
                FEATUREWORD_OS := FEATUREWORD_OS & 1 [0:1];

[OS]        SPPFEATURES_OS := EXCHANGE_FEATURES (WORDNUM_OS, FEATUREWORD_OS);

[SPP]       %Receives WORDNUM_OS and FEATUREWORD_OS from function call. Note that references
              % to these parameters use "<>"
              IF <WORDNUM_OS> is not a recognized feature word THEN
                FEATUREWORD_SPP := 0;
                GOTO CHECK_LAST;

FEATUREWORD_SPP := FEATURES_SPP[<WORDNUM_OS>];      %FEATURES_SPP: Array of feature word
                                                    % bit masks supported by SPP. This
                                                    % is hardcoded data.

              IF (FEATUREWORD_SPP NEQ <FEATUREWORD_OS>) THEN % Compare bits [47:46]
                IF a feature required by SPP is not supported by OS THEN
                  RESULT_SPP := FEATUREWORD_SPP & 1 [0:1]; % Set error bit in result
                  GOTO RETURN;
                % SUPPORTEDFEATURES_SPP: Array of supported features bit masks.
                SUPPORTEDFEATURES_SPP[<WORDNUM_OS>] := FEATUREWORD_SPP AND <FEATUREWORD_OS>;
              ELSE
                SUPPORTEDFEATURES_SPP[<WORDNUM_OS>] := FEATUREWORD_SPP;
```

FIG. 4A



```
CHECK_LAST:
IF Bit0 of <FEATUREWORD_OS> set AND did not receive all expected feature words THEN
  IF any of the remaining features are required by the SPP THEN
    RESULT_SPP := FEATUREWORD_SPP & 1 [0:1]; % Set error bit in result
    GOTO RETURN;
  ELSE
    % Set remaining words in SUPPORTEDFEATURES_SPP array to zero;
  ELSE
    RESULT_SPP := FEATUREWORD_SPP & 0 [0:1]; % Non-error Result (reset error bit)

RETURN:
  RETURN (RESULT_SPP);

[OS] IF Bit0 of SPPFEATURES_OS is set THEN
  %Fatal error. Abort system initialization. Report feature mismatch to
  %operations, etc. System Stopped.

IF (FEATUREWORD_OS NEQ SPPFEATURES_OS) THEN
  IF a feature required by OS is not supported by SPP THEN
    % Fatal error. Abort system initialization. Report feature mismatch to
    % operations, etc. System Stopped.
  ELSE
    % SUPPORTEDFEATURES_OS: Array of supported features bit masks.
    SUPPORTEDFEATURES_OS[WORDNUM_OS] := FEATUREWORD_OS AND SPPFEATURES_OS;
  ELSE
    SUPPORTEDFEATURES_OS[WORDNUM_OS] := FEATUREWORD_OS;

IF more feature words to exchange THEN
  BEGIN
    WORDNUM_OS := WORDNUM_OS + 1;
    GOTO LOOP;
  END;
```

FIG. 4B